Greedy Algorithms: Problem 1

We have n natural numbers, where n is an even number, which have to come together forming pairs of two numbers each. Then, from each pair, the sum of its two components is obtained, and from all these results the maximum is taken.

Design a greedy algorithm that creates the pairs so that the maximum value of the sums of the numbers of each pair is as small as possible, showing that the candidate selection function used provides an optimal solution.

Example: assuming that the data is in the following vector

5	8	1	4	7	9

Let's see a couple of ways to solve the problem (not necessarily the optimal one):

We select as a couple the consecutive elements

In this way we get the pairs (5, 8), (1, 4) and (7, 9); then, adding the components we have the values 13, 5 and 16, so the final result is 16.

We select as a couple the opposite elements in the vector

Now we have the pairs (5, 9), (8, 7) and (1, 4); adding we get 14, 15 and 5, so the final result is 15 (better than before).

Will there be a better result for this problem? Can a method be generalized that gives us a correct greedy algorithm for any amount of data, and that is independent of the value of those data?

Greedy Algorithms: Problem 1

Resolution Strategy

Different strategies

- Comparing the sum of consecutive values,

5	8	1	4	7	9

Cost = Max {(5+8), (1+4), (7+9)} = {15, 5, 16} = {16}

- Comparing the sum of opposed elements in the vector.

5	8	1	4	7	9

Cost = Max {(5+9), (8+7), (1+4)} = {14, 15, 5} = {15}

- Ordering the vector. Comparing the sum of consecutive values,

1	4	5	7	8	9

Cost = Max {(1+4), (5+7), (8+9)} = {5, 12, 17} = {17}

- Ordering the vector. Comparing the sum of opposed elements in the vector.

1	4	5	7	8	9

Cost = Max {(1+9), (4+8), (5+7)} = {10, 12, 12} = {12}

This method is general and independent of the data in the vector and, therefore, is correct.

Greedy Algorithms: Problem 1

Strategies

$$\begin{array}{|c|c|c|c|c|c|}\hline 5 & 8 & 1 & 4 & 7 & 9 \\ \hline\end{array} \longrightarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & 4 & 5 & 7 & 8 & 9 \\ \hline\end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 1+9 & 4+8 & 5+7 \\ \hline\end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 10 & \mathbf{12} & 12 \\ \hline\end{array}$$

- **Can a method be generalized to obtain a correct greedy algorithm to any amount of data and independent of their value?**

This methos is general for any amount of data and independent of their valiues: it could be integer or real data and the algorithm is still correct

Greedy Algorithms: Problem 1

Greedy Components for the Algorithm in Problem 1

1. **Set of candidates**: All the data.
2. **Set of Decisions**: Compare opposed data.
3. **Function that determines the solution of the problem**: Maximum of the compared opposed data.
4. **Completable**: Yes, with this strategy, you always have to cover all the vector.
5. **Selection Function**: Is the function which makes the implemented greedy algorithm is correct, working or not. The one ordering the vector and takes the higher of the partial sums of the compared opposed ,.
6. **Objective Function**: It is the one which returns the data off higher ,.

Algorithm Problem1 is Correct

Greedy Algorithms: Problem 1

```
const n = ...
type vector= array[1… n] of integer
type pairs= array[1… n%2] of integer
fun GroupPairs (I/O Input: vector ; O Output: pairs) ret Maximum: integer
var i: integer
     OrderIncrease ( Input )
     Maximum= -1
     for i=1  to  n%2 do
          Output [ i ] = Input  [ i ] + Input  [ n - i + 1]
          if  Output  [ i ] > Maximum then
               Maximum = Output [ i ] //Objective Function
          eif
     efor
     return Maximum
Efun
```

Selection Function